DTIC FILE COPY

ARO 24960.33-MA

(2)

AD-A212 160

ON IMPLICATIONAL DEPENDENC FAMILIES POSSESSING FINITE ARMSTRONG RELATIONS

KAZEM TAGHVA*

Department of Computer Science and Electrical Engineering

University of Nevada, Las Vegas
Las Vegas, Nevada 89154

89 9 11 120

UNCLASSIFIED MASTER COPY - FOR REPRODUCTION PURPOSES

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) ARO 24960.33-MA-REP
6a. NAME OF PERFORMING ORGANIZATION Univ. of Nevada 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION U. S. Army Research Office
6c. ADDRESS (City, State, and ZIP Code) Las Vegas, NV 89154	7b. ADDRESS (City, State, and ZIP Code) P. O. Box 12211 Research Triangle Park, NC 27709-2211
8a. NAME OF FUNDING/SPONSORING ORGANIZATION U. S. Army Research Office 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DAAL03-87-G-0004
8c. ADDRESS (City, State, and ZIP Code) P. O. Box 12211 Research Triangle Park, NC 27709-2211	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
On Implication Dependency Families Possessing Finite Armstrong Relations

12. PERSONAL AUTHOR(S)
Kazem Taghva

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM ____ TO ____	August 24, 1989	8

16. SUPPLEMENTARY NOTATION The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Data Dependencies, Functional Dependencies, Database Design, Armstrong Relations

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Let $X \neq \emptyset$ be a finite collection of nonempty relations over the relation scheme $R(A_1, A_2, ..., A_n)$; then the closure of X under embedding and direct product (up to isomorphism) is a *finitely generated Implicational Dependency family* (ID-family) generated by X. In this paper, we show that the class of finitely generated ID-families is identical to the class of those ID-families which possess a finite Armstrong relation.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code) / 22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

ON IMPLICATIONAL DEPENDENCY FAMILIES POSSESSING FINITE ARMSTRONG RELATIONS

KAZEM TAGHVA*
Department of Computer Science
The University of Nevada
Las Vegas, NV 89154

August 14, 1989

Abstract

Let $X \neq \emptyset$ be a finite collection of nonempty relations over the relation scheme $R(A_1, A_2, ..., A_n)$; then the closure of X under embedding and direct product (up to isomorphism) is a *finitely generated Implicational Dependency family* (ID-family) generated by X. In this paper, we show that the class of finitely generated ID-families is identical to the class of those ID-families which possess a finite Armstrong relation.

1 Introduction

Data dependencies such as *functional dependencies* (FDs), *multivalued dependencies* (MVDs), and *join dependencies* (JDs) have played an important role in the design of databases[2][3]. In addition, they have been used as integrity constraints in an integrity-checking mechanism[3]. The legal databases are

*This research was supported in part by U.S. Army Research Office under grant #DAAL03-87-G-0004.

those which obey the constraints specified by the database administrator originally. Consequently, we are interested in studying families of instances characterized by a given set of dependencies such as FDs, MVDs, etc.

The class of *Implicational Dependencies* (IDs) was defined by Fagin[2] as the logical generalization of the previously defined class of full dependencies. Properties of ID-families are mainly studied in [2],[4],[5],[7], in particular, it is shown that the collection of ID-families is closed under join and projection.

In[5], it is shown that a collection of relations over scheme $R(A_1, A_2, ..., A_n)$ is axiomatizable by IDs if and only if it contains a trivial database and it is domain independent and closed under embedding and direct products.

In this paper, we use the above result to establish that the collection of ID-families with a finite Armstrong relation and the collection of finitely generated ID-families are identical.

Vardi[8] has established a finite set of IDs with no finite Armstrong relation. This, together with the above result, implies that finitely specifiable ID-families are not finitely generated.

2 Preliminaries

In this paper, we assume readers to be familiar with [2], and [5]. We will follow the notation of [2]. In addition, throughout this paper we only deal with scheme $R(A_1, A_2, ..., A_n)$.

Following Fagin[2], we define an *Implicational Dependency* (ID) to be a typed sentence σ of the form $\forall x_1 \forall x_2 ... \forall x_m(\alpha_1 \wedge \alpha_2 ... \wedge \alpha_k \rightarrow \beta)$, where each α_i is an atomic formula of the form $R(y_1, y_2, ..., y_n)$ and β is an atomic formula of the form $R(y_1, y_2, ..., y_n)$ or $y_i = y_j$, where $y_d \in \{x_1, x_2, ..., x_m\}$. We also assume that $k \geq 1$ and each x_i occurs in some α_j. For example, the formula $\forall a \forall b \forall c_1 \forall c_2 \forall d_1 \forall d_2 R(a, b, c_1, d_1) \wedge R(a, b, c_2, d_2) \rightarrow c_1 = c_2)$ represents the FD $AB \rightarrow C$ for the 4-ary relation scheme $R(A, B, C, D)$, and the formula $\forall a \forall b_1 \forall b_2 \forall c_1 \forall c_2 R(a, b_1, c_1) \wedge R(a, b_2, c_2) \rightarrow R(a, b_1, c_2)$ represents the MVD $A \rightarrow\rightarrow B$ for the 3-ary relation scheme $R(A, B, C)$.

Let r and s be relations for R (our relations are all finite relations), then we define the direct product of r and s, in notation $r \times s$, to be the set of all tuples $t = ((t_{11}, t_{21}), (t_{12}, t_{22}), ..., (t_{1n}, t_{2n}))$ such that $t_1 = (t_{11}, t_{12}, ..., t_{1n}) \in r$ and $t_2 = (t_{21}, t_{22}, ..., t_{2n}) \in s$. For example, the direct product of the first two relations in the following diagram is the third relation.

r

A	B	C
a	b	c
$a\prime$	$b\prime$	$c\prime$

s

A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

$r \times s$

(a, a_1)	(b, b_1)	(c, c_1)
(a, a_2)	(b, b_2)	(c, c_2)
$(a\prime, a_1)$	$(b\prime, b_1)$	$(c\prime, c_1)$
$(a\prime, a_2)$	$(b\prime, b_2)$	$(c\prime, c_2)$

The direct product of $r_1 \times r_2 \times ... \times r_m$ is defined as usual. Also, we define $Dom(r)$ to be $Dom_r(A_1) \times Dom_r(A_2) \times ... \times Dom_r(A_n)$, where each $Dom_r(A_i)$ is the set of all the ith coordinates of r. For example, the $Dom(r)$ in the above diagram is:

$Dom(r)$

A	B	C
a	b	c
a	b	$c\prime$
a	$b\prime$	c
a	$b\prime$	$c\prime$
$a\prime$	b	c
$a\prime$	b	$c\prime$
$a\prime$	$b\prime$	c
$a\prime$	$b\prime$	$c\prime$

For the relation scheme $R(A_1, ..., A_n)$, we also assume a countably in-finite underlying domain for each A_i from which A_i takes its values. Let r and s be nonempty relations for R, then $f = (f_1, f_2, ..., f_n)$ is called an *embedding* from s to r if f_i is a 1-1 function from $Dom_s(A_i)$ to $Dom_r(A_i)$ for each i and for any tuple $(a_1, ..., a_n) \in Dom(s)$, then $(a_1, ..., a_n) \in s$ iff $(f_1(a_1), f_2(a_2), ..., f_n(a_n)) \in r$. In fact, embedding is a typed 1-1 homomorphism between two structures. In case such f exists, we say s can be *embedded* into r. An embedding f is called an *isomorphism* if f is onto. We will use the notation $r \cong s$ to show that r and s are isomorphic. A subset s of r is called a *substructure* of r if $Dom(s) \cap r = s$. It is obvious that if s is a substructure of r, then the identity map from $Dom(s)$ to $Dom(r)$ is an embedding.

Let Σ be a set of IDs, then $SAT(\Sigma)$ is the set of all finite relations satisfying Σ. A nonempty collection of relations F is an ID-family if there exists a set Σ of IDs such that $F = SAT(\Sigma)$. In case Σ is finite, we say F is *finitely specifiable* ID-family.

Let Σ be a set of IDs, then $\Sigma_* = \{\sigma \mid \Sigma \models \sigma\}$, i.e. Σ_* is the set of all IDs which logically follow from Σ. A relation r is called an *Armstrong relation* if all members of Σ_* are true in r and all other IDs are false in r. Armstrong relations and their applications are extensively studied in [1], [2], and [6].

For any collection K of relations, let

$$\text{SK} = \{\ \text{r} \mid \text{r can be embedded into some member of K}\}$$
$$\text{PK} = \{\ \text{r} \mid r \cong r_1 \times r_2 \times ... \times r_n \text{ for } r_i \text{ members of K}\}$$

The next theorem gives a characterization for ID-families.

Theorem 2.1 *[5]Let F be a family of relations for R, then F is an ID-family iff:*
(1) F is closed under P.
(2) F is closed under S.
(3) F contains a singleton.

We would like to mention here that Makowsky and Vardi[5] use the term "subdatabase" instead of "substructure".

3 Main Result

Let $X = \{r_1, r_2, ..., r_n\} \neq \emptyset$ be a collection of nonempty relations for R, then theorem 2.1 implies that SPX is an ID-family generated by X (note that condition (3) is trivially satisfied as any tuple t in some r_i will form the substructure $\{t\}$ for r_i). In case X contains a single relation, we will say SPX is *singly* generated.

The next two lemmas imply that the collection of finitely generated ID-families and the collection of singly generated ID-families are identical.

Lemma 3.1 *Let s_1 and s_2 be substructures of r_1 and r_2 respectively, then $s_1 \times s_2$ is a substructure of $r_1 \times r_2$.*

Proof. Straightforward.

Lemma 3.2 *Let $X = \{r_1, r_2, ..., r_m\}$ be a collection of nonempty relations for R, then $SPX = SP\{r_1 \times r_2 \times ... \times r_m\}$.*

Proof. Let $t_2, t_3, ..., t_m$ be tuples in $r_2, r_3, ..., r_m$ respectively. By lemma 3.1, $r_1 \times \{t_2\} \times ... \times \{t_m\}$ is a substructure of $r_1 \times r_2 \times ... \times r_m$. Now since r_1 is isomorphic to $r_1 \times \{t_2\} \times ... \times \{t_m\}$, it follows that r_1 is a member of $SP\{r_1 \times r_2 \times ... \times r_m\}$. Similarly, we can show $r_i \in SP\{r_1 \times r_2 \times ... \times r_m\}$ for $i = 2, 3, ..., m$.

We now establish a sequence of results to prove our main result.

Lemma 3.3 *Let $\{F_i \mid i \in I\}$ be a collection of ID-families, then $G = \bigcap\{F_i \mid i \in I\}$ is an ID-family.*

Proof. Since singleton relations satisfy all IDs, it is clear that $G \neq \emptyset$. To prove the lemma, we will use theorem 2.1. Let $r_1, r_2 \in G$, then $r_1, r_2 \in F_i$ for each i. Therefore, $r_1 \times r_2 \in F_i$ for each i. Hence, $r_1 \times r_2 \in G$ and G is closed under products. Similarly we can prove that G is closed under substructure.

Definition 3.1 *Let X be a collection of relations over R, then the smallest ID-family containing X is defined to be:*

$$G(X) = \bigcap\{F \mid X \subseteq F \text{ and F is an ID-family }\}$$

Lemma 3.3 together with the fact that $X \subseteq SAT(\emptyset)$ implies that $G(X)$ always exists.

Theorem 3.1 *Let $X = \{r\}$, then SPX is an ID-family and r is an Armstrong relation.*

Proof. Since SPX is closed under S and P, then by theorem 2.1, $SPX = SAT(\Sigma)$ for some set of IDs Σ. The definition of G(X), smallest ID-family containing X, and theorem 2.1 together imply that SPX = G(X).

Let $\Gamma = \{\gamma \mid r \models \gamma$ and γ is an ID $\}$, then by definition of G(X), we have $SPX \subseteq SAT(\Gamma)$. Also, since every member of Σ is true in r, we have $\Sigma \subseteq \Gamma$ which implies $SAT(\Gamma) \subseteq SAT(\Sigma)$. This shows that

$$G(X) = SPX = SAT(\Gamma) = SAT(\Sigma)$$

Now we show that r is an Armstrong relation for Σ. It is obvious that any σ which is the logical consequence of Σ is true in r. Suppose σ is not the logical consequence of Σ, then there exists a relation $s \in SAT(\Sigma)$ such that σ is false in s. Now, if σ is true in r, then σ will be a member of Γ. But this is a contradiction since $s \in SAT(\Sigma) = SAT(\Gamma)$.

Finally, we show that the collection of finitely generated ID-families is the same as the collection of ID-families possessing a finite Armstrong relation.

Theorem 3.2 *The collection of finitely generated ID-families and the collection of ID-families possessing a finite Armstrong relation are identical.*

Proof. By theorem 3.1 and lemma 3.2, finitely generated ID-families possess finite Armstrong relations. On the other hand, suppose F possesses a finite Armstrong relation r. Since $SP\{r\} = SAT(\Sigma)$ is the smallest ID-family containing r, it follows that $SAT(\Sigma) \subseteq F$. Now let $s \in F$ and suppose s is not a member of $SAT(\Sigma)$, then there exists a $\sigma \in \Sigma$ which is false in s. Since r is an Armstrong relation for F, it follows that σ is false in r. But this is a contradiction as $r \in SAT(\Sigma)$. This shows that $F \subseteq SAT(\Sigma)$.

4 Final remarks

Let $r = \{t\}$, then $F = SP\{r\}$ is the collection of all singletons together with $\emptyset$. F can be axiomatized by the set of all IDs. In addition, F can be axiomatized by the following finite set of IDs:

$$\forall x_1 ... \forall x_n \forall y_1 ... \forall y_n (R(x_1, x_2, ..., x_n) \wedge R(y_1, y_2, ..., y_n) \rightarrow x_1 = y_1)$$
$$\forall x_1 ... \forall x_n \forall y_1 ... \forall y_n (R(x_1, x_2, ..., x_n) \wedge R(y_1, y_2, ..., y_n) \rightarrow x_2 = y_2)$$
...
...
...
...
$$\forall x_1 ... \forall x_n \forall y_1 ... \forall y_n (R(x_1, x_2, ..., x_n) \wedge R(y_1, y_2, ..., y_n) \rightarrow x_n = y_n)$$

This example motivates one to investigate the relationship between finitely generated and finitely specifiable ID-families. Vardi[8] has constructed a finite set of IDs with no finite Armstrong relation. This together with theorem 3.2 shows that finitely specifiable ID-families are not finitely generated. We do not know whether finitely generated ID-families are finitely specifiable.

References

[1] C. Beeri, M. Dowd, R. Fagin, and R. Statman, On the structure of Armstrong relations for functional dependencies. *J. ACM* 31,1(1984) pp. 30-46.

[2] R. Fagin, Horn clauses and database dependencies, *J.ACM* 29,4(Oct. 1982), pp. 952-985.

[3] M.M. Hammer and D.J. Mcleod, Semantic integrity in a relational data base system. In *Proc. of 1st Int. Conf. on Very Large Databases* (Sept. 1975), pp. 25-47.

[4] R. Hull, Finitely specifiable implication dependency families. *J. ACM* 31,2(Apr. 1984), pp. 210-226.

[5] J. Makowsky and M. Vardi, On the expressive power of data dependencies. *Acta Informatica* 23,3(1986), pp. 231-244.

[6] H. Mannila and K. Raiha. Automatic generation of test data for relational queries. *J. of Computer and System Sciences* 38, 2(1989), pp. 240-258.

[7] K. Taghva, Some characterizations of finitely specifiable implicational dependency families, Information Processing Letters 23(Oct. 1986) pp. 153-158.

[8] M.Y. Vardi, Personal communication.